Chapter 13
Approximate Semantic Event Processing in Real-time Linked Dataspaces

Souleiman Hasan and Edward Curry

Keywords Event processing · Semantic matching · Best-effort · Internet of things · Dataspaces

13.1 Introduction

Within dataspaces, data sources are not necessarily fully integrated or homogeneous in their schematics and semantics. For dataspaces to support a real-time response to situations of interest when a set of events take place, for example from sensor readings, there is a need for a principled approach to tackling data heterogeneity within real-time data processing. In this chapter, we detail techniques for developing dataspace support services for dealing with the semantic heterogeneity of real-time data.

In the following sections, we build upon the discussion so far in this book and focus on best-effort semantic matching for real-time data processing in dataspaces (Sect. 13.2). In Sect. 13.3, we discuss an approximate semantic matching service for real-time data processing in dataspaces, represented by the approximate semantic and thematic event processing models. The elements of the approach are detailed in Sect. 13.4. Then, in Sect. 13.5, we discuss an instantiation of these event processing models, with their evaluation in Sect. 13.6. We finish with an analysis of related work in Sect. 13.7, and then conclude the chapter in Sect. 13.8.

13.2 Approximate Event Matching in Real-time Linked Dataspaces

Real-time data sources are increasingly forming a significant portion of the data generated in dataspaces. This in part is due to increased adoption of the Internet of Things (IoT) and the use of sensors for improved data collection and monitoring of

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_13

daily activities in smart buildings, smart energy, smart cities, and others. In this section, we explore the concepts of dataspaces and event processing to understand data management challenges in dataspaces with real-time data sources. We then introduce the approximate semantic event matching services to address the challenge of semantic matching of heterogeneous events. The design of the service is based on our existing work in approximate semantic event matching, including [151, 272, 307, 308], which is brought together in this chapter and contextualised for use within the dataspace paradigm.

13.2.1 Real-time Linked Dataspaces

Driven by the adoption of the IoT, smart environments are enabling data-driven intelligent systems that are transforming our everyday world, from the digitisation of traditional infrastructure (smart energy, water, and mobility), the revolution of industrial sectors (smart autonomous cyber-physical systems, autonomous vehicles, and Industry 4.0), to changes in how our society operates (smart government and cities). To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among systems. A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. The dataspace paradigm pushes the boundaries of traditional data management approaches in two main dimensions [2]: *Administrative Proximity*, which describes how data sources within a space of interest are close or far in terms of control; and *Semantic Integration*, which refers to the degree to which the data schemas within the data management system are matched up. These dimensions form part of the three boundaries (knowledge, value, and ecosystem from the Knowledge Value Ecosystem (KVE) Framework) that need to be crossed in order for knowledge exchange to occur among systems within a data ecosystem (see Chap. 2 for further discussion on this topic).

Within dataspaces, data sources are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required,

it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. In order to understand the requirements of real-time data processing, we will explore the event processing paradigm [139].

13.2.2 Event Processing

In event processing, data items that are shared within the dataspace in real time are called *events*. Data sources which produce events are called *event producers*. Users and software which are interested in an event, or set of events, are called *event consumers*. For example, in a smart building, there can be IoT-based data sources which produce information continuously, such as energy consumption sensors and motion detection sensors within an office. Data items produced by such sensors are the events. The sensor is the event producer. A building manager may be interested in situations where a light in an office is left on while the office is unoccupied. In this example, the building manager and the software representation of their interest (the event query) would be the event consumer.

Thus, an essential part of the event processing paradigm is the matching mechanism between the events and the interests of event consumers. This is similar to the concept of query processing in relational database management systems, where events replace the concept of a data tuple, and subscriptions or rules replace the concept of queries. In a specific family of event processing systems, called stream processing, queries take the name of continuous queries as they are evaluated continuously against data.

In terms of crossing system boundaries for data sharing, the decoupled nature of event-based systems reduces their administrative proximity. However, in terms of semantic integration, event-based systems currently require tight semantic integration [151]. Acknowledging that the challenges to dataspaces such as loose administrative proximity and loose semantic integration can face real-time data sources, the question becomes: how can we support the loose administrative proximity and semantic coupling within real-time data processing? To answer this question, we look no further than the literature of the event processing paradigm itself. A core principle in event processing is decoupling, which refers to the lack of explicit agreements in order to increase scalability as defined by Eugster et al. [142]. Three

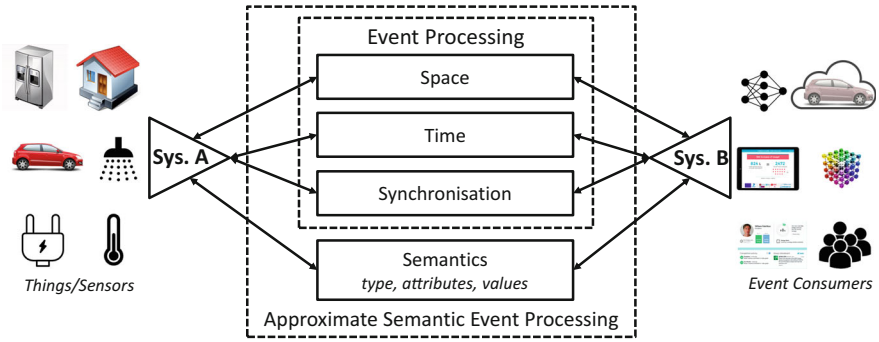

Fig. 13.1 The decoupling dimensions of event processing (Adapted from [151])

main dimensions have been recognised in the event processing literature concerning decoupling, as illustrated in Fig. 13.1:

- *Space Decoupling*: Which means that event producers and consumers do not hold addresses, such as IPs, of each other.
- *Time Decoupling*: Which means that event producers and consumers do not have to be active at the same time.
- *Synchronisation Decoupling*: Which means that event producers and consumers do not block each other when exchanging events.

We build on the concept of decoupling to meet the principles of Real-time Linked Dataspaces as detailed in Chap. 4, that is, an event processing paradigm that supports many formats of data, does not depend on schema agreement, and supports a best-effort approximate and pay-as-you-go approach. We identify this as a new dimension for event processing systems which we call *loose semantic coupling*.

13.3 The Approximate Semantic Matching Service

Loose coupling of event processing systems on the semantic dimension reflects a low cost to define and maintain rules concerning the use of terms, and a low cost for building and agreeing on the event semantic model. This requirement forms the foundation of our semantic matching models and their enabling elements, as discussed in the remainder of this chapter.

13.3.1 Pay-As-You-Go Service Levels

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers

defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The semantic matching models have been used within an approximate semantic event processing support service within the RLD. The functionality of the service follows the 5 Star pay-as-you-go model (detailed in Chap. 4) of the RLD. The approximation semantic matching service has the following tiered-levels of support:

1 Star **No Service:** No semantic matching is supported.
2 Stars **Semantic Matching:** Approximate semantic matching at the attribute-value of events and subscriptions.
3 Stars **Thematic Matching:** Thematic matching of events with the use of theme tags to more accurately describe events in a low-cost manner.
4 Stars **Entity-Centric:** Event matching is performed over entity-centric event graphs (e.g. RDF).
5 Stars **Context-Aware:** Context-aware semantic matching of events with the use of external knowledge from the dataspace.

13.3.2 Semantic Matching Models

The main semantic matching models and elements of the approach are presented, with respect to the event flow model presented by Cugola and Margara in [139]. The core components of an event engine in their model are the event *Receiver*, the *Decider*, the *Producer*, and the event *Forwarder*. Event *Sources*, *Consumers*, and *Users* interact with the engine through protocols and condition/action *Rules*. Figure 13.2 presents an elaboration of Cugola and Margara's model with additional models and elements for approximate semantic event processing. Two models form the basis for the approach: (I) the approximate event matching model, and (II) the thematic event matching model, which are outlined in the following sections.

13.3.3 Model I: The Approximate Event Matching Model

This model extends the current event processing paradigm through the following:

- Event processing rules are equipped with the *tilde* $\sim$ semantic approximation operator so users can express their delegation to the event engine to match similar, or related, event terms, to the term used in a subscription. The background semantic model for approximation is a statistical model built from co-occurrences of terms in a large corpus of plain text documents. For instance, the following subscription tells the event engine to match it to events generated with device equal to *'laptop'* or similar terms and to match the value "room 112" with the term *'office'* or related terms such as *'room'* or *'zone'*.

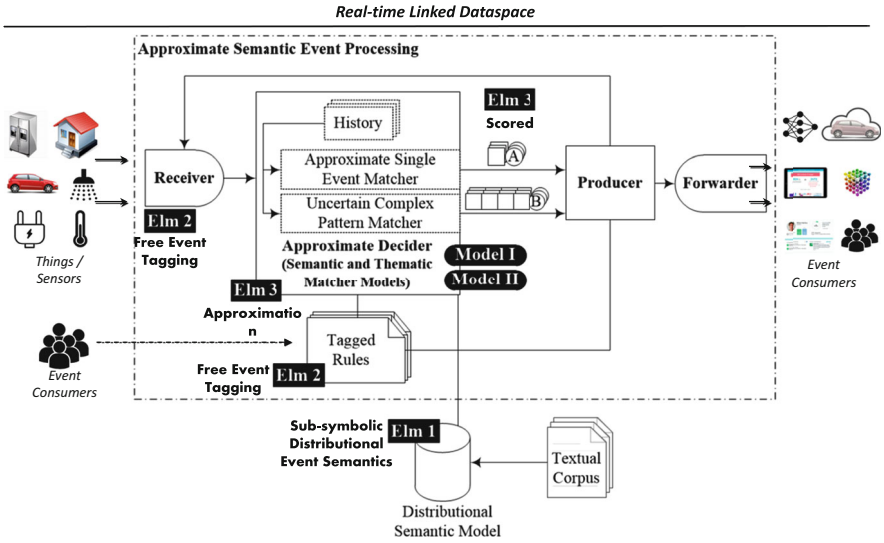

Fig. 13.2 Models and elements in the approximate semantic event processing model

$$\{\textbf{type} = \text{increased energy usage event},$$
$$\textbf{device} = \text{laptop} \sim,$$
$$\textbf{office} \sim= \text{room 112}\}$$

- The single event matcher is equipped with matching and mapping algorithms to detect events semantically relevant to approximate subscriptions. For instance, let an event of *increased energy consumption* be represented as follows:

$$\{\textbf{type} : \text{increased energy consumption event},$$
$$\textbf{measurement unit} : \text{kilowatt-hour},$$
$$\textbf{device} : \text{computer},$$
$$\textbf{office} : \text{room 112}\}$$

The most probable mapping, or the top-1 mapping, of this event to the previous subscription is generated as a probable scored result. It can be described as follows:

$$\begin{aligned}\sigma^* = \{&(\textbf{type} = \textbf{increased energy consumption event}\\ &\leftrightarrow \text{type : increased energy usage event}),\\ &(\textbf{device} \sim= \textbf{laptop} \sim\leftrightarrow \text{device : computer}),\\ &(\textbf{office} = \textbf{room 112} \leftrightarrow \text{office : room 112})\}\end{aligned}$$

- The complex pattern matcher can then perform probabilistic reasoning to deduce the probabilities of occurrences of the derived events in the action parts of the complex rules.

13.3.4 Model II: The Thematic Event Matching Model

This model suggests associating free tags that describe the themes of types, attributes, and values in events and subscriptions, in order to clarify their meanings. For instance, the previous *increased energy consumption* event is associated with tags as follows:

$$\{\text{appliances}, \text{building}\}$$

These tags help disambiguate the meaning of terms in the event such as *'energy'* and *'office'* and move them closer to the energy management domain in smart buildings. Thematic events can more easily cross semantic boundaries as (1) they free users from needing a prior semantic top-down agreement, and (2) they carry approximations of events' meanings composed of payloads and theme tags which, when combined, carry less semantic ambiguities. An approximate matcher exploits the associated thematic tags to improve the quality of its uncertain matching of events and subscriptions.

13.4 Elements for Approximate Semantic Matching of Events

The approach can be conceptually decomposed into three main elements, as outlined in the following sections and illustrated in Fig. 13.2.

13.4.1 Elm 1: Sub-symbolic Distributional Event Semantics

This element stems from the need for loosening the semantic coupling between event producers and consumers. If semantic coupling can be quantified by the number of mappings between symbols, that is, terms, and meanings, then a semantic model that condenses these mappings can be particularly useful. Ontological models require granular agreements on the symbol-meaning mappings, which is proportional to the number of symbols. However, distributional vector space semantics leverage the statistics of terms co-occurrence in a large corpus to establish semantics [309]. For instance, the terms *'power'* and *'electricity'* would frequently co-appear in an energy management domain corpus. Thus, they can be assumed to be related, and this can be leveraged within energy event matching. Using such a model leaves event producers and consumers to loosely agree on the corpus as a representative of their common knowledge and decrease the need for granular agreements on every individual term of the domain.

13.4.2 Elm 2: Free Event Tagging

This element stems from the need to enable event processing within a loosely coupled model to effectively and efficiently allow users to adapt the conveyed events' meanings in different domains and situations. Free tagging of events and subscriptions do not introduce any coupling components between participants, in contrast to the case of top-down fixed taxonomies. This element builds on the success of free tagging, known as folksonomies, within social media research [310]. For instance, the term *'energy'* when used in an event tagged by the tags {*'building'*, *'appliance'*} helps the matcher distinguish the meaning of *'energy'* and associate it with the domain of power management, rather than associating it with the domain of sport or diet, for example. When used to process IoT events, free tagging can create "Thingsonomies" as a way to support the discovery and use of events from Things [308].

13.4.3 Elm 3: Approximation

This element stems from the realisation that loosening the coupling between event producers and consumers at the semantic and pragmatic levels introduces uncertainties to the engine. Uncertainty results from not exactly knowing which event's tuples shall be mapped to which subscription's tuples. For instance, with the loose agreements on terms' semantics, there are various possible mappings between an event and a subscription such as:

$$\sigma_1 = \{(\textbf{device} = \textbf{laptop} \leftrightarrow \text{device} : \text{computer}),\\ (\textbf{room} = \textbf{room 112} \leftrightarrow \text{office} : \text{room 112})\}$$

$$\sigma_2 = \{(\textbf{device} = \textbf{laptop} \leftrightarrow \text{office} : \text{room 112}),\\ (\textbf{room} = \textbf{room 112} \leftrightarrow \text{device} : \text{computer})\}$$

Each mapping has a different probability which reflects the uncertainty of the matching. Approximation at the core of the event processing engine can tackle uncertainties and complement the elements mentioned earlier.

13.4.4 Elements Within the Event Flow Functional Model

The main elements of the approach can be unified and placed into the event processing functional model illustrated in Fig. 13.2 as follows:

- *Elm1—Sub-symbolic Distributional Event Semantics*: The actual distributional semantic model could be built outside of the event processing engine by indexing a textual corpus. The resulting model forms the basis to compare any two strings in events and subscriptions, as they get decoded into their vector representations. Vectors form the basis for distance and similarity measures.
- *Elm2—Free Event Tagging*: Events are flowing from event sources, and subscriptions get tagged by users before they are considered for matching. Users use free tags to enhance events and subscriptions and improve their interpretation by the matcher.
- *Elm3—Approximation*: Events are matched in the decider against subscriptions. The decider is now approximate, and the result of matching is represented by scored events (Elm3-*Scored*) which signify their relevance to each subscription. The decider makes use of the semantic model and the tags when matching the events.

13.5 Instantiation

The instantiation of the models requires a concrete model for the events, the subscription language, and the matching model, as discussed in the following sections.

13.5.1 Events

The most elaborate event model is the instantiation of the thematic event model, which is based on an attribute-value model. Each event is a pair of sets: a set of theme tags and a set of tuples. Each theme tag is a single word or a multi-word term. Each tuple consists of an attribute-value pair. No two distinct tuples can have the same attribute. An example energy consumption event is represented as follows:

$$
\begin{aligned}
&(\{\textbf{energy}, \textbf{appliances}, \textbf{building}\},\\
&\{\textbf{type} : \text{increased energy consumption event},\\
&\textbf{measurement unit} : \text{kilowatt-hour},\\
&\textbf{device} : \text{computer},\\
&\textbf{office} : \text{room 112}\})
\end{aligned}
$$

The formal definition of the event model is as follows: let E be the set of all events, let TH be the set of all possible theme tags, and let A and V be the sets of possible attributes and values respectively. Let AV be the set of possible attribute-value pairs, that is, tuples, such that $AV = \{(a, v): a \in A \wedge v \in V\}$. An event $e \in E$ is a pair (th, av) such that $th \subseteq TH$ and $av \subseteq AV$ are the set of theme tags and the set of tuples, respectively.

13.5.2 Subscriptions

Each subscription is a pair of two sets: a set of theme tags and a set of conjunctive attribute-value predicates. Each theme tag is a single word or a multi-word term. Each predicate uses the equality operator to signify exact equality or the tilde operator for approximate equality when indicated. Other Boolean and numeric operators such as $!=$, $>$, and $<$ are kept out of the language for the sake of discourse simplicity. Each predicate consists of an attribute, a value, and specifications of the semantic approximation for the attribute and the value. The most notable feature of the language is the *tilde* $\sim$ operator that helps specify the approximation for an attribute/value when it follows it. An example subscription to energy usage events is as follows:

$$
\begin{aligned}
&(\{\textbf{power}, \textbf{computers}\},\\
&\{\textbf{type} = \text{increased energy usage event} \sim ,\\
&\textbf{device} \sim = \text{laptop} \sim ,\\
&\textbf{office} = \text{room 112}\})
\end{aligned}
$$

The author of the subscription specifies that the device can be a *'laptop'*, or something related semantically to *'laptop'*. The subscription also states that the attribute *'device'* itself can be semantically relaxed. However, it states that the event's *'office'* must be exactly *'room 112'*.

The formal definition of the language model is as follows: let S be the set of subscriptions, let TH be the set of all possible theme tags, and let A and V be the sets of possible attributes and values, respectively, which can be used in a subscription. Typically, there are no restrictions on A or V, and the user is free to use any term or combination of terms. Each predicate is a quadruple which consists of the attribute, the value, and whether or not the attribute/value is approximated. Let P be the set of possible predicates, thus $P = \{p\colon p = (a, v, appa, appv) \in A \times V \times \{0, 1\}^2\}$. A subscription $s \in S$ is a pair (th, pr) where $th \subseteq TH$ and $pr \subseteq P$ are the set of theme tags and the set of predicates, respectively. The degree of approximation is the proportion of relaxed attributes and values. An exact subscription has 0% degree of approximation.

13.5.3 Matching

The matching model is illustrated in Fig. 13.3. An approximate semantic single event matcher M decides on the semantic relevance, or mapping, between a subscription s and an event e based on the semantic mapping between attribute-value predicates of s and attribute-value tuples of e. The model is detailed in [151, 272].

An example mapping between the event and the approximate subscription described above is as follows:

$$\sigma = \{(\textbf{type} = \textbf{increased energy consumption event} \leftrightarrow \text{type : increased energy usage event}),$$
$$(\textbf{device} \sim= \textbf{laptop} \sim\leftrightarrow \text{device : computer}),$$
$$(\textbf{office} = \textbf{room 112} \leftrightarrow \text{office : room 112})\}$$

M works in two modes: the top-1 mode that decides on the most probable mapping between s and e, and the top-k mode which decides on the top-k probable mappings to be used later for complex event processing.

The formal definition of matching is as follows: let $C = s \times e$ be the set of all possible correspondences between predicates of s and tuples of e. $\forall c = (p, t) \in C \Rightarrow p \in s \wedge t \in e$. $\Sigma = 2^C$ is the power set of C and represents all the possible mappings between s and e. There are exactly n correspondences in any valid mapping σ where n is the number of predicates in the subscription s.

For any valid mapping σ, a probability function quantifies the probability of every predicate-tuple correspondence $(p, t) \in \sigma$ such as (device = laptop$\sim \leftrightarrow$ device: computer). There also exists a probability function that quantifies the probability of the overall mapping σ, among other possible mappings. Both functions form probability spaces P_σ and P.

In the basic approximate semantic matching model, the semantic relatedness is directly calculated from vector representations of terms as suggested by the distributional semantic model. In the thematic model, probabilities are calculated based on the combined similarity matrix that is based on the thematic pairwise attributes or values semantic relatedness scores. Thematic semantic relatedness measure uses the tags to project and adjust the vector representations of words in a parametric vector space model before calculating their similarity as illustrated in Fig. 13.3 and detailed in [272].

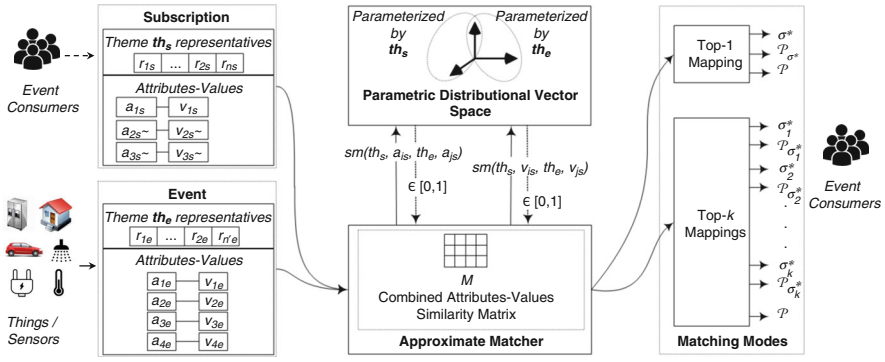

Fig. 13.3 The matching model (Adapted from [272])

13.6 Evaluation and Discussion

To evaluate the models, an evaluation event set of 50,000 events has been semantically expanded out of seed event sets from actual deployments of IoT intelligent systems for smart city, energy management, building, and relevant datasets, to evaluate the approximate semantic event matching model as detailed in [151]. Similarly, 14,743 events were generated to evaluate the thematic event matching model as detailed in [272].

Evaluation metrics can be classified into two categories: effectiveness and efficiency metrics [311]. Effectiveness metrics measure the quality of event matching. A fundamental requirement is the existence of a ground truth which divides events into relevant and irrelevant with respect to each approximate subscription. *Precision*, *Recall*, and combined $F_1 Score$ have been used for effectiveness evaluation. The metric used for evaluating time efficiency is the matcher's *Throughput* defined as $Throughput = (Number\ of\ processed\ events)/(Time\ unit)$.

Additionally, to measure the loosening in the semantic coupling, we use two measures: *alternative number of exact subscription rules* that would be needed in a semantically coupled model, and the *degree of approximation* used in the approximate subscriptions. In the thematic model, the *number of tags* used is also considered. These measures are compared to the exact matching model's numbers, which would typically have many exact subscription rules that have zero degrees of approximation because of the direct coupling.

13.6.1 Evaluation of the Approximate Semantic Event Matching Model

The efficiency evaluation aims to compare the throughput of the approximate semantic matching model against an exact matching model based on query rewriting. Given a set of approximate subscriptions, each approximate subscription can be rewritten as a set of conjunctive statements in the Esper event engine, each of which is a set of attribute-value pairs resulting by replacing the approximate parts of a subscription with related terms from the WordNet dictionary. The ground truth's thesaurus is Merriam-Webster [312].

The experiment was conducted with ten sets of $10-100$ approximate subscriptions of 50% degree of approximation using Explicit Semantic Analysis (*ESA*) as a semantic relatedness [313] measure. Figure 13.4 shows that the approximate matcher delivers 94–97% matching quality, which is higher than the 89–92% delivered by the WordNet-based rewriting approach equipped with exact matching. The rewriting approach outperforms the approximate model in throughput when the pairwise semantic relatedness scores are calculated at run-time. However, the approximate matcher based on pre-computed *esa* scores outperforms in throughput with around 91,000 events/s compared to around 19,100 events/s on average.

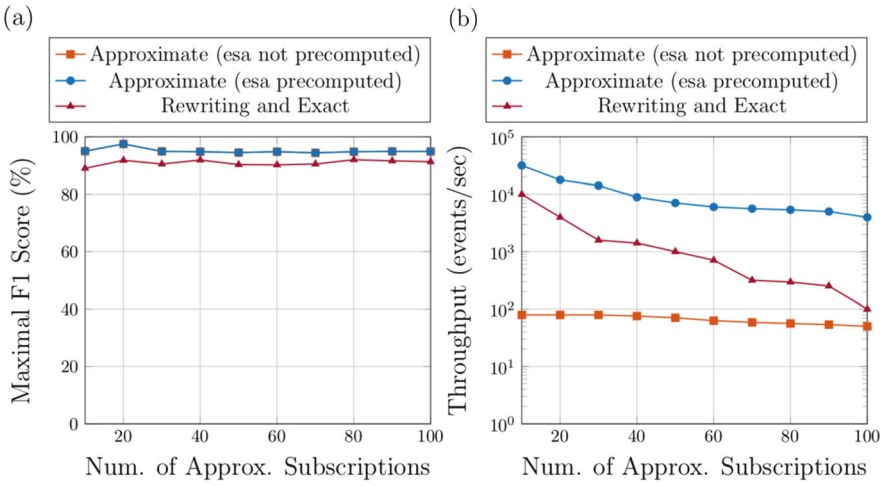

Fig. 13.4 The approximate semantic event matcher results. (**a**) Effectiveness. (**b**) Efficiency [151]

Finally, to achieve the 100% of F_1Score and the throughput of an exact matcher, there is a need to write manually all the possible rules that are equivalent to the approximate rules. To quantify this situation, we measure how many exact rules are required to compensate for approximate rules given that the rewriting is done with the ground truth thesaurus, which is Merriam-Webster. This showed that about 74,000 exact rules are needed to cover all events compared to a maximum of only 100 rules for the approximate matcher. Thus, the exact matcher is a non-feasible solution in highly semantically heterogeneous environments.

13.6.2 Evaluation of the Thematic Event Matching Model

To evaluate the thematic approach, we compare it with non-thematic approximate semantic event processing described above. A large event set is generated with a specific theme as well as a set of subscriptions which assume no semantic agreements and 100% degree of approximation. The thematic matcher is compared with the non-thematic matcher when various theme tags are used.

The baseline matcher achieves 62% of F_1Score and a throughput of 202 events/s averaged over five runs which represent its worst case due to full approximation of the subscription by using the $\sim$ operator on all subscription's predicates.

Each cell in Fig. 13.5 represents the average F_1Score of the sample of five sub-experiments, each of which uses a different combination of events and subscriptions themes tags. For instance, the sub-experiments of the cell in the second column and tenth row from the bottom left, all use two terms to describe the theme of an event, and ten terms to describe a subscriptions theme, and the event theme terms set is a subset of the subscription theme terms set.

Figure 13.5 shows that thematic matching outperforms the non-thematic matching in F_1Score for more than 70% of combinations with scores 62–85% and an average of 71% versus 62% for the baseline. Thematic matching performs worse when the number of thematic tags is very small, for example, using just one term as a theme tag. The performance is worst in the bottom triangular half of the figure with F1Score widely ranging from 4% to 62%. Larger themes for subscriptions quickly improve the effectiveness as opposed to the opposite effect by event themes. This reflects the asymmetric relationship between the many heterogeneous events versus fewer subscriptions. Thus, more terms are needed in subscription themes to discriminate relevant events.

Figure 13.6 shows the average throughput for each combination of events and subscriptions theme tags. It suggests that the thematic approach outperforms the non-thematic matcher for more than 92% of the sub-experiments, with a throughput of 202–838 and an average of 320 versus 202 events/s. The improved throughput is due to the thematic filtering of the space during the thematic projection phase, which saves time during the semantic relatedness calculation.

The results show that the thematic approach is limited when users can provide only a small number of tags for subscriptions, and when hard real-time deadlines are required. Otherwise, the results suggest that the use of fewer terms to describe events, around 2–7, and more to describe subscriptions, around 2–15, can achieve a good matching quality and throughput together with low error rates. This is concentrated in the middle to the upper left side of Figs. 13.5 and 13.6. The

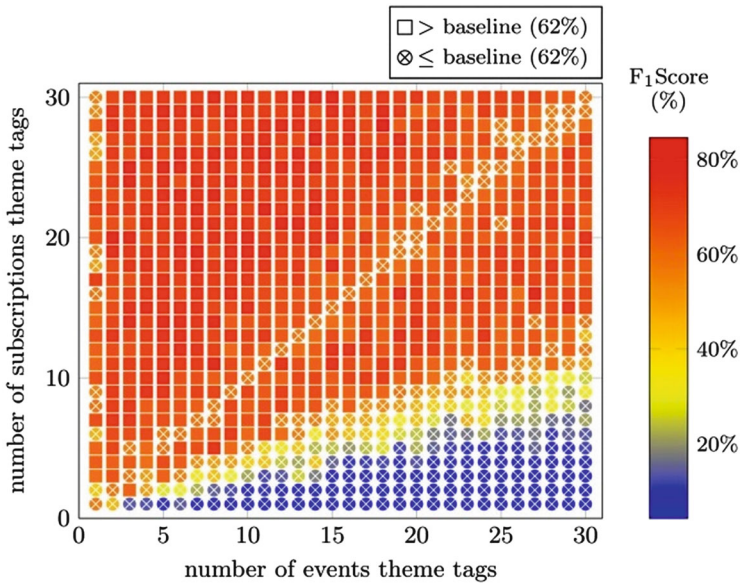

Fig. 13.5 Effectiveness evaluation of the thematic event matcher [272]

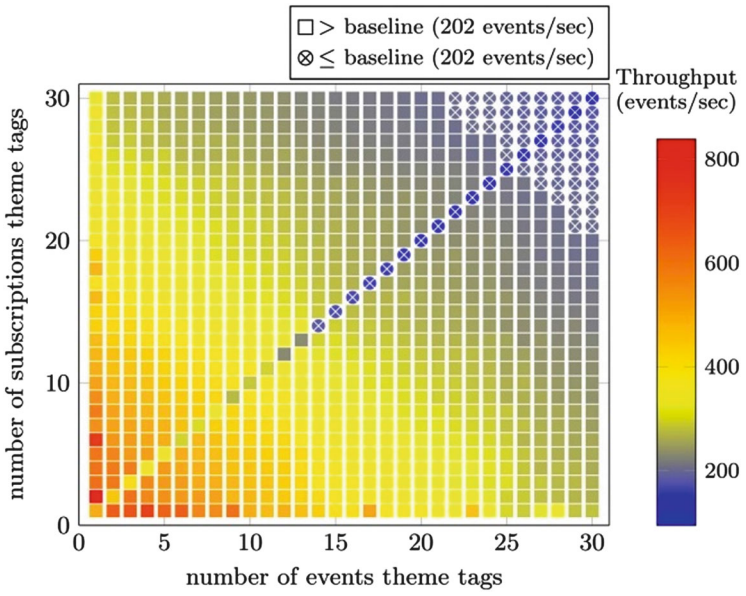

Fig. 13.6 Efficiency evaluation of the thematic event matcher [272]

evaluation data indicates that events and subscriptions need to be associated with only a few thematic tags.

13.7 State-of-the-Art Analysis

The event processing literature related to approximate semantic event matching can be classified into five major classes:

- *Content-Based Event Processing*: In content-based event processing, event sources and consumers use the same event types, attributes, and values without any additional description of meaning external to the rules and events. The principal works in this category are those by Carzaniga et al. [314] (SIENA), Eugster et al. [315], and Fiege et al. [316] (Rebeca). Such approaches are effective with the timely matching and routing of events, but they assume an implicit agreement on the semantics of events outside of the event engine, which is a type of semantic coupling that does not scale in heterogeneous environments.
- *Concept-Based Event Processing*: In this category, participants can use different terms and values and still expect matchers to be able to match them correctly thanks to explicit knowledge representations such as thesauri and ontologies that encode semantic relationships between terms. The principal works in this category are those by Petrovic et al. [317] (S-ToPSS), Wang et al. [318] (OPS), Zeng and Lei [319], and Blair et al. [320] (CONNECT). Given agreements on explicit

models, efficient and effective detection of positive and negative matchings can be achieved. Nonetheless, agreements on explicit models may become an unfeasible task to achieve due to high levels of heterogeneity at large scales.

- *Approximate Event Processing*: Approaches in this category are distinguished by a matching model that is not Boolean. The principal works in this category are those by Zhang and Ye [321] (FOMatch), Liu and Jacobsen [322, 323] (A-TOPSS), and Wasserkrug et al. [324]. These approaches reduce semantic coupling due to their ability to deal with the uncertainties of users about semantics. Time efficiency is high, but effectiveness is lower due to the approximate model, which allows some false-positive/-negatives to occur.
- *Query-Based Fusion*: Approaches in this category adopt declarative languages similar to SQL. These languages support operators of semantics like relational join. They enable semantic description and matching of events as well as the fusion of streams of events with background context data. The principal works in this category are those by Arasu et al. [325] (CQL), Teymourian et al. [326], Le-Phuoc et al. [298] (CQELS), and Anicic et al. [299] (EP-SPARQL). These approaches are like concept-based models in that they assume an explicit model of semantics which might be hard to agree on.
- *Semantic and Context Transformation*: Approaches in this category handle events individually and perform a set of transformations on them to move from one semantic model to another. The principal works in this category are those by Freudenreich et al. [327] (ACTrESS), and Cilia et al. [328, 329] (CREAM). These approaches consider semantics to have one nature and impact on event matching. They are effective and efficient in matching. Nonetheless, semantic models that depend on ontologies and conversion functions require agreements which form a coupled mode that limits scalability in heterogeneous environments.

The literature analysis shows that related approaches are mainly based on symbolic semantics, exact matching, and ad hoc domain specificity, which generally requires agreements that are difficult to achieve in highly distributed and open environments such as smart environments.

13.8 Summary and Future Work

This chapter discusses the approximate semantic event processing model in answer to the requirement of loose semantic coupling, which is necessary to adopt the principles of dataspaces to real-time data. We found that to meet this requirement, the event processing paradigm needs to be enhanced with additional elements such as: sub-symbolic distributional semantics, free event tagging, and approximation. We showed that these elements could transform the event matcher into an approximate semantic matcher with a probabilistic model. The resulting approximate and thematic matchers provide an effective matching quality and efficient time

performance, and most importantly, they require minimal upfront agreements among event producers and consumers on event semantics.

Future directions would include the development of the approximate matcher to encompass new event and subscription models beyond attribute-value models, and the extension of subscription languages with numeric operators such as less-than and greater-than operators. An interesting direction is the extension of the thematic matching model to other unstructured event types such as images and videos, which would imply opportunities for new intelligent applications in real-time dataspaces of unstructured data [30].